

Distilled Branch Predictors

Simha Sethumadhavan

Department of Computer Science, Columbia University
simha@columbia.edu

Abstract—We study a branch prediction organization that places a small single cycle “student” predictor before a larger, more accurate predictor. Unlike a hybrid/ensemble predictor, the student is trained on the larger, slower predictor’s direction, instead of each predictor being trained on resolved branch outcomes. This makes the smaller predictor a low latency approximation of the larger predictor. We also evaluate an optional bypass to avoid the large predictor accesses altogether for branches whose resolved outcomes have stayed in one direction. When using a 32KiB, two cycle TAGE predictor as a teacher and a 4KiB single cycle predictor as a student, teacher/student training improves VFS by 0.0132 on hard32 and 0.0138 on full168 over an independently trained same size bimodal P1. When using a 3 cycle CBP2025 winner as teacher, the student configuration with the bypass for one sided branches improves VFS by 42.8% on hard32 and 45.2% on full168 over Seznec’s CBP 2025 submission.

Index Terms—branch prediction, teacher/student learning, TAGE, frontend latency, energy efficiency.

I. INTRODUCTION

Combining multiple branch predictions has a long history. Smith’s pioneering work made clear that no single low cost rule captures all branch behavior [1]. Yeh and Patt then showed the power of two level adaptive history, where recent branch outcomes select among learned patterns rather than using only a PC indexed counter [2]. McFarling’s 1993 Combining Branch Predictors paper made the hybrid predictor explicit: a chooser predictor is used to select one among many predictors that has recently worked best for that branch [3].

There are several distinct reasons to combine predictors: accuracy, because local, global, loop, path, and perceptron like predictors see different patterns [3], [4]; alias control, because different indexes and histories create different interference patterns; frontend bandwidth, because multiple branch and multiple block predictors help wide machines fetch more instructions per cycle [12]; latency tolerance, because fast/slow and override predictors can redirect only when a later predictor is worth the delay [15]; and energy or area control, because confidence, filter, and sharing schemes avoid expensive accesses on easy branches or reuse component state [13].

This paper belongs to the latency and energy part of these lineages, but changes the training target of one of the predictors. Traditional two level or hybrid designs train each component independently to predict the resolved branch outcome, then choose among those independently trained predictions. We instead train the fast predictor (the student predictor, hereon P1) to approximate the slower, larger (teacher, or P2) predictor’s output. When the student provides the same prediction as the teacher, the frontend obtains the correct

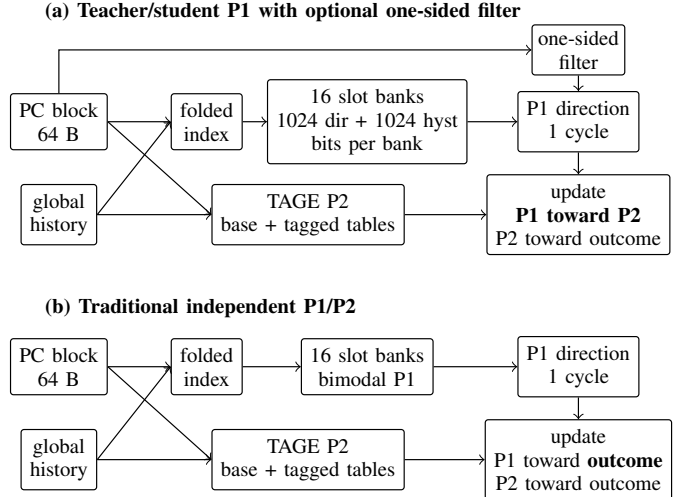


Fig. 1. One cycle P1 organizations. The teacher/student design trains P1 toward the P2 predictor’s output; the traditional two level design trains P1 directly on resolved branch outcomes. The PC-local one-sided is an optional bypass that can suppress a P2 access for branches that remain directionally one-sided for long latency P2s.

decision sooner. When the student disagrees, the teacher still supplies the correction and provides the student with a training example that can be updated quickly. Essentially, the student model distills the state used by the teacher to make the predictions, and since the training is preformed in an online manner, it also captures prediction locality. In our evaluation, using this training strategy improves VFS relative to an independently trained P1 for every TAGE and hashed-perceptron P2 configuration, while experiments with Seznec’s CBP2025 show one boundary case in which a branch outcome trained bimodal P1 is competitive with teacher/student predictor.

II. TEACHER/STUDENT OPERATION

Figure 1(a) shows the proposed organization and Table I summarizes the access behavior. P1 is sized to be a one cycle direction predictor. For experiments that use TAGE as P2, P1 is organized as 16 independent slot banks, one bank for each possible four byte instruction position in a 64 byte fetch block. Each bank contains 1024 direction bits and 1024 hysteresis bits. The read index is the fetch block address folded with six bits of recent global direction history, and returns one direction bit for each slot.

P2 is the predictor whose output defines the student’s training target. Unless otherwise stated, for this submission, P2 is the geometric history TAGE predictor listed in Table II;

TABLE I
STUDENT ACCESS, PREDICTION, AND UPDATE BEHAVIOR.

Mechanism	Microarchitectural behavior
P1 student read	Read one direction bit and one hysteresis bit from the selected slot bank. The bank index is the 64 byte line address folded with six bits of recent global direction history.
P2 teacher read	Read the teacher predictor unless an optional one directional bypass predictor suppresses the access. TAGE uses its base table, tagged geometric history tables, and alternate prediction; the CBP2025 comparison uses the ensemble in Seznec’s predictor structures.
Prediction use	P1 supplies the first cycle direction. If P2 is accessed and later disagrees, P2 supplies the correcting direction and the disagreement contributes to the late prediction terms in VFS.
Student update	Train P1 toward P2 prediction. Agreement strengthens hysteresis; disagreement weakens hysteresis and rewrites the direction bit only when the entry is weak. For the student used with Seznec CBP 2025 please consult text. On a mispredicted bypassed P2 prediction, remove the entry from the bypass filter.
Teacher update	Train P2 on the resolved branch outcome on every non bypassed prediction and on recovery after a bypassed misprediction.
Optional one sided reset	Track a tagged PC local direction table. A bypassed miss trains P2 on recovery and resets that row before future bypasses are allowed.

the experiments also evaluate a hashed perceptron P2 and the Seznec CBP2025 predictor. We assume that P2 is pipelined and that a new lookup may enter every cycle, while the prediction becomes available after its two or three cycle latency depending on the predictor configuration. Unless an optional bypass suppresses the lookup, P2 is accessed for each prediction request and is trained on the resolved branch outcome.

A key distinction from a conventional ensemble predictor is how P1 updates are performed. A traditional P1 (say bimodal) ensemble, shown in Fig. 1(b), trains its counters directly on resolved outcomes. The student instead trains toward P2’s direction. When P1 and P2 agree, the hysteresis bit is strengthened. When they disagree, hysteresis is weakened first. Thus, another teacher disagreement moves the entry to the teacher’s current label.

We also model one sided branch detection as an optional filter for P2. The mechanism adds a small PC-tagged structure that tracks whether recent resolved outcomes for a branch have remained one-sided. After training, that state may provide the final direction and suppress the P2 access for the current prediction. If a skipped prediction is wrong, P2 is consulted and trained on recovery, and the PC-local entry corresponding to that PC is reset so the branch must relearn its one-sided phase before it can bypass P2 again. We evaluate this option

with the CBP2025 teacher only because the late predictor is three cycles and the benefits with two cycle predictors will likely be minimal because most of the table access for P2 happens in the first cycle.

III. RESULTS

All results are based on 1M warmup branches and 1M measured branches per trace. The hard32 suite is the 32 trace hard subset while full168 is the full trace suite provided for the competition. Table II lists the TAGE teacher parameters: number of tagged tables (Tables), entries per tagged table (Tag ent), base table entries (Base ent), tag width (Tag), and maximum history length (Hist). The history lengths for accessing the tables are geometrically scaled. All results use a 64 byte prediction block with 16 possible 4 byte instruction slots, and the same fixed one cycle student folded with six bits of global history. The larger student used in the Seznec CBP2025 experiments, s10, is 1024 entry table. The table is indexed by the PC and global history as before. Unlike the TAGE student, this student, includes a tag to avoid aliasing, and the tag is hash of the PC and global/local history. There is an optional confidence predictor that is incremented on taken and decremented on not taken. There is also a one sided direction bit. The same update rule still applies as the TAGE student. The independently trained P1 used for baseline comparisons uses the same 32Kib state budget and the same 16 per-slot banks as the TAGE-sweep student, but trains that one cycle predictor directly on resolved branch outcomes. P2 is also trained independently from the outcome, and P2 supplies the final direction when P1 and P2 disagree.

The TAGE 4x P2 configuration has eight tagged geometric history tables, 2048 entries per tagged table, 11 bit tags (4 for slot and 7 for hash of PC and folded history), a 4096 entry bimodal base table, and a 100 branch maximum history (the 100 history was chosen somewhat arbitrarily based on geometric scaling on 1.5 per table). The sizes were picked to fit under 2 clock cycles. The P2 lookup is issued for every prediction in the TAGE experiments and updated on every resolved conditional branch, so the later predictor continues to receive the training examples needed to correct future predictions. The CBP2025 comparisons use the larger S10 student described above.

The teacher predictor is assumed to be pipelined to enable a new lookup can enter each cycle, and older lookups advance through the stages. A nonpipelined multicyle P2 would accumulate extra requests behind a one cycle frontend and would eventually stall and hence is not evaluated.

Table III shows the measured TAGE and hashed perceptron results. Within this sweep, the TAGE 4x configuration has the best VFS on both suites. TAGE 8x and TAGE 16x continue to reduce MPKI but lose value to extra energy and, for TAGE 16x, a three cycle P2. The hashed perceptron is competitive in MPKI on hard32 but too energy heavy to win VFS.

TABLE II
TAGE TEACHER PARAMETERS. ALL ROWS USE A 64 BYTE PREDICTION BLOCK, 16 INSTRUCTION SLOTS, AND THE FIXED 32K BIT STUDENT.

P2	Tables	Tag ent.	Base ent.	Tag	Hist.
1x	8	512	1024	10	64
2x	8	1024	2048	10	80
4x	8	2048	4096	11	100
8x	10	2048	4096	11	128
16x	12	2048	4096	12	192

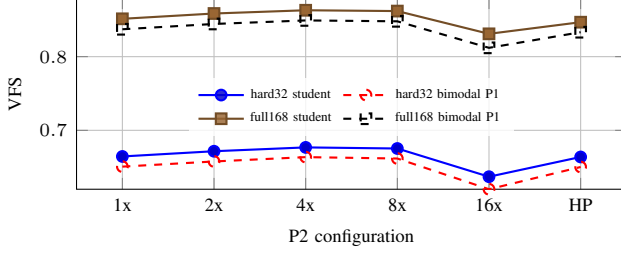


Fig. 2. Teacher/student training outperforms an independently trained same size bimodal P1 across the TAGE and hashed-perceptron P2 configurations. HP is the hashed perceptron P2.

The above table shows results of holding P1 size, P1 latency, and P2 fixed. The student combination is better all these configurations. On the TAGE 4x configuration, teacher/student training improves VFS by 0.0132 on hard32 and 0.0138 on full168 relative to the independently trained bimodal P1.

Larger P2 Predictors To study the scalability of the teacher student model we experiment with larger P2 predictors. The large TAGE reference keeps the same 64 byte prediction block as the earlier 2 cycle TAGE configurations, but expands the late predictor to 20 tagged tables, 2048 entries per tagged table, a 65536 entry base table, 13 bit tags, and a 1024 branch maximum history. The CBP2025 comparison uses the real Seznec CBP2025 reference predictor for direction and update behavior and charges explicit timed accesses for its TAGE/SC structures.

For our teacher/student scheme to be viable, the P2 accuracy gain per added latency and energy, discounted by student/teacher disagreement, must be high. In the TAGE sweep, growing from 4x to 16x reduces MPKI by only 0.147 on hard32 and 0.129 on full168, which is below the break even point even under the two cycle sensitivity model. With an effective first cycle predictor, CBP2025 reduces MPKI enough to overcome the three cycle P2 latency. The student with the optional one sided reset improves VFS by 42.8% on hard32 and 45.2% on full168 over the three cycle CBP2025 winner alone. The independently trained bimodal P1 with the same CBP2025 P2, improves VFS by 42.7% on hard32 and 46.8% on full168.

IV. RELATED WORK

We can think of branch predictors as hardware implemented online supervised learning systems. With this view, one can look at machine learning literature for analogues. One of the

first works to talk about model distillation — the process of compacting a model for time and space optimization — was by Bucilua, Caruana, and Niculescu-Mizil (2006) who showed that a compact model can be trained from the outputs of a larger ensemble [5]. Hinton et al. in 2015 popularized the term knowledge distillation by training a student on a teacher’s softened output distribution [6], and later work extended the idea to intermediate representations [7], online or mutual distillation among peers [8], [9], and large language model compression [10], [11]. Our use follows the model compression intuition but differs in mechanism. Unlike ML works, the student is *not* trained offline from a transfer set or with a gradient loss. It is an *online*, hardware constrained supervised learner that observes the dynamic branch stream and updates a one cycle table towards a larger branch predictor.

Patent US8959320B2 is the closest related work to ours. This patent from Apple, issued in 2015, places a fast predictor before a later, more accurate predictor and uses the later predictor to control training of the fast structure [14]. The patent focusses on preventing the student predictor from being trained on alternating branches. It is unclear why this special mechanism is necessary because simply folding the global history with the PC (as we do in our student) would prevent this pathology. The need for such special suppression could be microarchitecture specific, and may be explained if the mechanism were targeted at a predictor where the frontend infrastructure does not provide global histories but it unclear what specific microarchitectural decisions imposes this constraint. Nevertheless, we modeled the training (prevention) policy used in the patent in our framework and it performed worse than the comparable TAGE 4x student described before in terms of IPC and VFS score. It is entirely possible that purpose of the predictor is not direction prediction (alone) and that this is not a fair comparison. We provide the above only for completeness.

V. CONCLUSION

In this paper we presented a technique for distilling/approximating a larger branch predictor with a smaller branch predictor for latency and energy savings. The key change is very simple: instead of the training the smaller, faster predictor on resolved branch outcomes we train it to mimic the output of the larger, slower predictor. As with all teacher/student setups, as long as non conflicting labels in the teacher map to the same student entry, the student can faithfully follow the teacher, and this will provide the speed and energy benefits of the student, and will approach the accuracy of the teacher.

The experiments show that distillation enabled by the teacher student setup improves VFS scores by cutting down on prediction and misprediction latencies while maintaining accuracy. When a TAGE predictor is used as a teacher, the VFS gains are modest around 2% but for a larger slower, highly accurate predictor like the Seznec’s CBP 2025 entry that uses a TAGE with additional ensemble predictors, the gains are substantial around 40%. There is much to be explored in

TABLE III

MEASURED RESULTS AND TRADITIONAL BIMODAL P1 COMPARISON. THE BIMODAL P1 HAS THE SAME ONE CYCLE LATENCY, 16 BANK ORGANIZATION, AND 32K BIT BUDGET AS THE STUDENT, BUT IS TRAINED ON RESOLVED BRANCH OUTCOMES.

Suite / P2	MPKI	EPI	IPC	P1/P2	Stud. VFS	Bimod. VFS	Gain
hard32 TAGE 1x	14.095	905	5.170	1/2	0.6645	0.6507	0.0138
hard32 TAGE 2x	13.500	1129	5.189	1/2	0.6716	0.6577	0.0139
hard32 TAGE 4x	13.003	1346	5.197	1/2	0.6768	0.6636	0.0132
hard32 TAGE 8x	12.924	1518	5.197	1/2	0.6753	0.6617	0.0136
hard32 TAGE 16x	12.856	1891	4.995	1/3	0.6370	0.6200	0.0170
hard32 hashed perc.	12.784	2001	5.092	1/2	0.6638	0.6502	0.0136
full168 TAGE 1x	6.515	856	5.705	1/2	0.8514	0.8372	0.0142
full168 TAGE 2x	6.085	1068	5.717	1/2	0.8586	0.8444	0.0142
full168 TAGE 4x	5.764	1276	5.722	1/2	0.8630	0.8492	0.0138
full168 TAGE 8x	5.690	1441	5.720	1/2	0.8619	0.8480	0.0139
full168 TAGE 16x	5.635	1799	5.527	1/3	0.8310	0.8120	0.0190
full168 hashed perc.	5.844	1910	5.662	1/2	0.8468	0.8332	0.0136

TABLE IV

VFS BREAKDOWN FOR THE SUBMITTED TAGE 4X CONFIGURATION. WPI IS WRONG PATH INSTRUCTIONS PER CORRECT PATH INSTRUCTION.

Component	hard32	full168
IPC input	5.1973	5.7222
CPI input	0.1300	0.0576
EPI input (fJ)	1346.5	1276.2
MPI	0.0130	0.0058
DPI	0.0076	0.0059
PPI	0.1598	0.1542
P1/P2 cycles	1/2	1/2
WPI	0.6758	0.3299
Speed term	0.4854	0.6734
Energy term	0.2010	0.3498
VFS	0.6768	0.8630

TABLE V

P2 GROWTH AND CBP2025 COMPARISON. TAGE SENSITIVITY ROWS RECOMPUTE VFS FROM THE SAME PER-TRACE COUNTS WITH THE LISTED P2 AVAILABILITY LATENCY. THE BIMODAL+CBP2025 ROW TRAINS THE ONE CYCLE P1 INDEPENDENTLY ON OUTCOMES AND USES CBP2025 WHEN P1 AND P2 DISAGREE; THE COMPARISON CHARGES P1 READS AND DIRECTION-BIT REWRITE CYCLES. THE STUDENT+CBP2025+RESET ROW ENABLES THE OPTIONAL ONE-SIDED BYPASS FROM TABLE I. IN P2-ONLY ROWS, THE LEFT SIDE OF P1/P2 REPORTS THE EARLIEST MODELED DIRECTION AVAILABILITY RATHER THAN A SEPARATE P1 PREDICTOR.

Configuration	P1/P2	hard32	full168	Note
large TAGE ref.	2/3	0.4567	–	P2-only
stud.+TAGE 4x	1/2	0.6768	0.8630	measured
stud.+TAGE 4x	1/3	0.6434	0.8369	latency
stud.+TAGE 16x	1/2	0.6704	0.8573	latency
stud.+TAGE 16x	1/3	0.6370	0.8310	measured
CBP2025 winner	–/3	0.5474	0.6573	P2-only
bimod.+CBP2025	1/3	0.7812	0.9646	outcome P1
stud.+CBP2025+reset	1/3	0.7818	0.9547	student+skip

predictors could enable even higher bandwidth front ends and more energy efficient microarchitectures.

Acknowledgements: An AI assistant (Codex from OpenAI) helped inspect and generate code, write scripts to summarize runs, prepare tables and plots, and edit the $\text{\LaTeX}$; the author made the research decisions and final claims. The author likes to thank Anita Simha for comments on the draft, and the organizers for the infrastructure created for the competition.

REFERENCES

- [1] J. E. Smith, “A study of branch prediction strategies,” ISCA, 1981.
- [2] T.-Y. Yeh and Y. N. Patt, “Two-level adaptive training branch prediction,” MICRO, 1991.
- [3] S. McFarling, “Combining branch predictors,” DEC WRL Technical Note TN-36, 1993.
- [4] D. A. Jimenez and C. Lin, “Dynamic branch prediction with perceptrons,” HPCA, 2001.
- [5] C. Bucilua, R. Caruana, and A. Niculescu-Mizil, “Model compression,” KDD, 2006.
- [6] G. Hinton, O. Vinyals, and J. Dean, “Distilling the knowledge in a neural network,” arXiv:1503.02531, 2015.
- [7] A. Romero, N. Ballas, S. E. Kahou, A. Chassang, C. Gatta, and Y. Bengio, “FitNets: Hints for thin deep nets,” ICLR, 2015.
- [8] Y. Zhang, T. Xiang, T. M. Hospedales, and H. Lu, “Deep mutual learning,” CVPR, 2018.
- [9] D. Chen, J.-P. Mei, C. Wang, Y. Feng, and C. Chen, “Online knowledge distillation with diverse peers,” AAAI, 2020.
- [10] V. Sanh, L. Debut, J. Chaumond, and T. Wolf, “DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter,” arXiv:1910.01108, 2019.
- [11] X. Xu, M. Li, C. Tao, T. Shen, R. Cheng, J. Li, C. Xu, D. Tao, and T. Zhou, “A survey on knowledge distillation of large language models,” arXiv:2402.13116, 2024.
- [12] A. Seznec, S. Jourdan, P. Sainrat, and P. Michaud, “Multiple-block ahead branch predictors,” ASPLOS, 1996.
- [13] H. H. J. Hum and S. J. Jourdan, “Sharing information to reduce redundancy in hybrid branch prediction,” US Patent 7,203,825, 2007.
- [14] J. Beaumont-Smith and R. B. Gunna, “Branch prediction using multiple branch predictors,” US Patent 8,959,320 B2, 2015.
- [15] T. Krishnamurthy and A. Jarvis, “Combined level 1 and level 2 branch predictor,” US Patent 8,788,797 B2, 2014.

terms of branch predictor distillation: Machine Learning works shows that letting multiple cheap student predictors train one another or train a leader student could be beneficial. Such